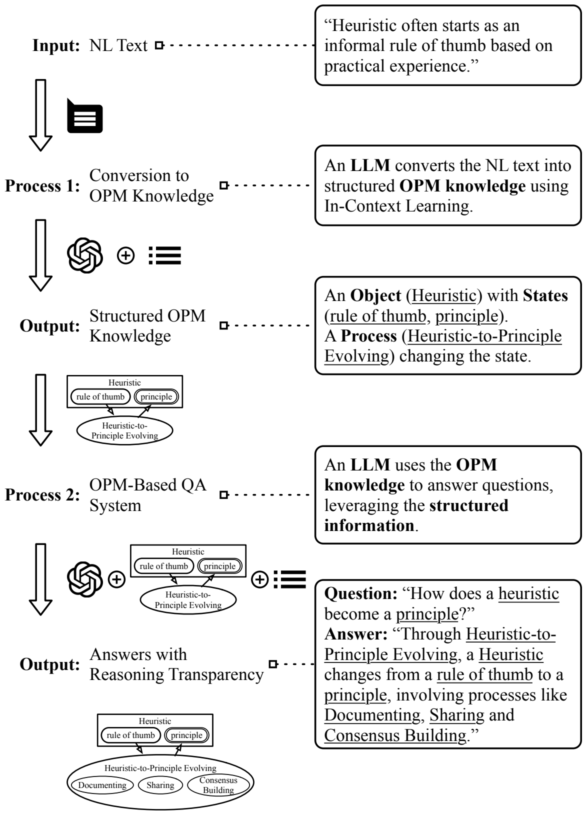
Neuro-Conceptual Artificial Intelligence: Integrating OPM with Deep Learning to Enhance Question Answering Quality

Xin Kang[1], **Veronika Shteingardt**[2], **Yuhan Wang**[1], **and Dov Dori**[2]
[1]Tokushima University, Tokushima, Japan
[2]Technion – Israel Institute of Technology, Haifa, Israel
kang-xin@is.tokushima-u.ac.jp, veronika-s@campus.technion.ac.il,
c612494013@tokushima-u.ac.jp, dori@technion.ac.il

Abstract

Knowledge representation and reasoning are critical challenges in Artificial Intelligence (AI), particularly in integrating neural and symbolic approaches to achieve explainable and transparent AI systems. Traditional knowledge representation methods often fall short of capturing complex processes and state changes. We introduce **Neuro-Conceptual Artificial Intelligence (NCAI)**, a specialization of the neuro-symbolic AI approach that integrates conceptual modeling using Object-Process Methodology (OPM) ISO 19450:2024 with deep learning to enhance question-answering (QA) quality. By converting natural language text into OPM models using in-context learning, NCAI leverages the expressive power of OPM to represent complex OPM elements—processes, objects, and states—beyond what traditional triplet-based knowledge graphs can easily capture. This rich structured knowledge representation improves reasoning transparency and answer accuracy in an OPM-QA system. We further propose transparency evaluation metrics to quantitatively measure how faithfully the predicted reasoning aligns with OPM-based conceptual logic. Our experiments demonstrate that NCAI outperforms traditional methods, highlighting its potential for advancing neuro-symbolic AI by providing rich knowledge representations, measurable transparency, and improved reasoning.


Input: NL Text
Process 1: Conversion to OPM Knowledge
Output: Structured OPM Knowledge
Heuristic
rule of thumb
principle
Heuristic-to-Principle Evolving
Process 2: OPM-Based QA System
Heuristic
rule of thumb
principle
Heuristic-to-Principle Evolving
Output: Answers with Reasoning Transparency
Heuristic
rule of thumb
principle
Heuristic-to-Principle Evolving
Documenting
Sharing
Consensus Building

"Heuristic often starts as an informal rule of thumb based on practical experience."

An **LLM** converts the NL text into structured **OPM knowledge** using In-Context Learning.

An **Object** (Heuristic) with **States** (rule of thumb, principle).
A **Process** (Heuristic-to-Principle Evolving) changing the state.

An **LLM** uses the **OPM knowledge** to answer questions, leveraging the **structured information**.

Question: "How does a heuristic become a principle?"
Answer: "Through Heuristic-to-Principle Evolving, a Heuristic changes from a rule of thumb to a principle, involving processes like Documenting, Sharing and Consensus Building."

Figure 1: Overview of the NCAI framework, illustrating how the LLM converts natural language text into structured OPM knowledge and uses it in OPM-QA for transparent reasoning. Starting from the text "Heuristic often starts as an informal rule of thumb . . . ", the model generates an OPM model and answers questions by referencing processes like *Heuristic-to-Principle Evolving*.

1 Introduction

Integrating neural and symbolic approaches in AI seeks to combine the learning capabilities of neural networks with the interpretability of symbolic reasoning (Besold et al., 2017; Garcez and Lamb, 2023). However, traditional knowledge representations, such as triplet-based knowledge graphs, are limited in capturing complex processes, state changes, and hierarchical relationships inherent in dynamic systems (Wang et al., 2017; Heinzerling and Inui, 2021; Shi et al., 2021). Additionally, neural networks are often viewed as a black box due to their opaque decision-making processes, which poses significant challenges in domains requiring transparent reasoning, such as healthcare and finance (Lipton, 2018; Rudin, 2019; Doshi-Velez and Kim, 2017; Tjoa and Guan, 2020).

Recent advancements have focused on enhancing AI reasoning capabilities by integrating language models with external knowledge sources.

Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ COLING 2025, pages 71–85
January 20th, 2025. ©2025 NeusymBridge Workshop

For example, combining language models with knowledge graphs has been applied to improve question answering systems (Yasunaga et al., 2021; Oguz et al., 2022; Shi et al., 2021; Zhang et al., 2023). Despite these efforts, fully capturing dynamic behaviors and providing transparent reasoning paths remains a challenge.

To address these limitations, we introduce **Neuro-Conceptual Artificial Intelligence (NCAI)**, a specialization of the neuro-symbolic AI approach, in which the symbolic component is an Object-Process Methodology (OPM ISO 19450:2024) conceptual model. OPM is a conceptual modeling language and methodology that unifies the system's structural and behavioral aspects within a single model (Dori, 2002; Dori et al., 2016). It represents objects (things that exist) and processes (things that transform objects) in both graphical and textual modalities. By combining OPM with the large language model (LLM), NCAI enhances reasoning transparency and answer accuracy in QA tasks.

An overview of the NCAI framework is illustrated in Figure 1. The framework begins by converting natural language text into structured OPM knowledge using in-context learning with an LLM. This structured knowledge is then used in an OPM-QA, which leverages the expressive power of OPM to represent complex processes and state changes that traditional triplet-based knowledge graphs cannot easily capture. By integrating conceptual modeling with deep learning, NCAI creates a pipeline that transforms unstructured text into a rich knowledge representation, enabling more effective AI reasoning and interpretability.

Our contributions in this work are threefold:

(1) We propose NCAI, which integrates OPM with deep learning to enhance reasoning transparency and answer accuracy.

(2) We develop OPM-QA that utilizes OPM knowledge to improve question-answering quality.

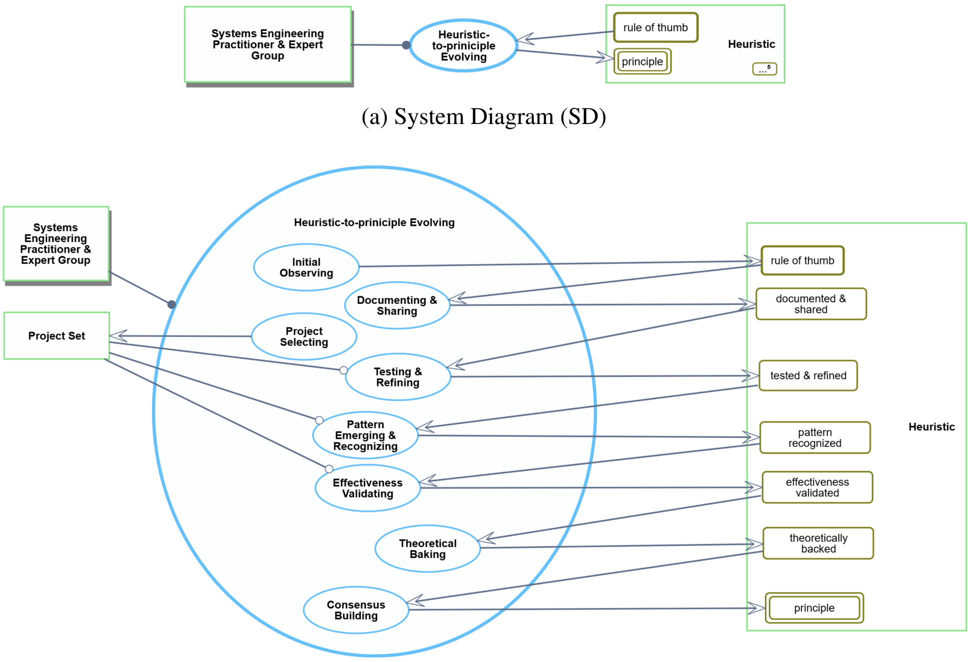
(3) We introduce transparency evaluation metrics to quantitatively assess how faithfully the predicted reasoning aligns with the conceptual logic defined by OPM, and we demonstrate the effectiveness of NCAI through experiments showing improved performance over traditional methods.

2 Related Work

Neuro-Symbolic AI Approaches Neuro-symbolic AI integrates neural networks with symbolic reasoning to harness the strengths of both paradigms (Besold et al., 2017; Garcez and Lamb, 2023). Challenges in achieving reasoning transparency and interpretability persist, with approaches such as symbolic knowledge distillation (West et al., 2022) and factual knowledge editing (De Cao et al., 2021) addressing these issues. Frameworks like TransferNet (Shi et al., 2021) and interpretable reasoning models for dialogue generation (Yang et al., 2022) aim to provide clear reasoning paths. In sentiment analysis and mental health, neuro-symbolic frameworks like TAM-SenticNet (Dou and Kang, 2024) and causal inference models (Ding et al., 2024b,a) enhance explainability and logical inference. Specifically, in aspect-based sentiment analysis (ABSA), models such as the Multi-Agent Collaboration (MAC) (Kang et al., 2024) and approaches to improve AI transparency using generative agents (Kang, 2024) demonstrate the potential of neuro-symbolic AI in providing transparent and rational sentiment analysis.

Interpretability and Transparency in Language Models Ensuring transparency and interpretability in AI decision-making is critical, particularly in complex systems (Lipton, 2018; Rudin, 2019). Various methods have been developed to enhance the interpretability of language models, including representation dissimilarity measures (Brown et al., 2023), SHAP-based explanation techniques (Mosca et al., 2022), and prompt-based explainers like PromptExplainer (Feng et al., 2024). Evaluation benchmarks for interpretability (Wang et al., 2022) and approaches to improve faithfulness and robustness (El Zini and Awad, 2022; Horovicz and Goldshmidt, 2024; Zhao et al., 2024) further contribute to making language models more transparent. Despite these advancements, achieving full transparency remains challenging, especially in applications requiring a clear understanding of the reasoning process.

Language Models and Knowledge Graphs for Question Answering Integrating language models with knowledge graphs has been a significant focus to enhance QA capabilities. Approaches like QA-GNN (Yasunaga et al., 2021), DRLK (Zhang et al., 2022), and UniK-QA (Oguz et al., 2022) combine language models with graph neural networks and dynamic interactions to improve reasoning in QA tasks. Frameworks such as CIKQA (Zhang et al., 2023) and Triple-R (Kanaani et al., 2024)



(a) System Diagram (SD)

(b) In-Zoomed Diagram (SD1)

Figure 2: Constructed OPDs illustrating the transformation of a *Heuristic* from a *rule of thumb* to a *principle* through various OPM elements—processes, objects, and states—within the OPM framework.

emphasize the integration of external knowledge sources for more accurate and interpretable reasoning. Additionally, methods like TaPERA (Zhao et al., 2024) enhance faithfulness and interpretability in long-form table QA through content planning and execution-based reasoning. These integrations, while improving performance, often involve complex architectures and still face challenges in achieving complete reasoning transparency.

3 NCAI Framework

3.1 Object-Process Methodology for NCAI

OPM unifies objects and processes within a single model, representing structural and behavioral aspects in both graphical and textual forms (Dori, 2002; Dori et al., 2016). OPM's bimodal property provides Object-Process Diagram (OPD) and Object-Process Language (OPL), enhancing understanding and reasoning transparency.

To illustrate OPM's capabilities, we use a running example based on natural language text describing the evolution of a *Heuristic* from a *rule of thumb* to a *principle*. This text serves as the input to the NCAI framework, as shown in Figure 1, and is provided in Appendix A.

Using this text, we constructed OPDs representing the processes and state changes of a *Heuristic*. The diagrams can be created and visualized using the OPCloud software (Dori et al., 2018; Kohen and Dori, 2021).

Figure 2 presents the constructed OPDs. The System Diagram (SD) in Figure 2a captures the overall transformation of object *Heuristic* from state *rule of thumb* to state *principle* through the process *Heuristic-to-Principle Evolving*. The In-Zoomed Diagram (SD1) in Figure 2b provides a detailed view of the subprocesses involved, such as *Documenting & Sharing*, *Testing & Refining*, *Pattern Emerging & Recognizing*, *Effectiveness Validating*, *Theoretical Backing*, and *Consensus Building*.

The corresponding OPL for the System Diagram (SD) and the In-Zoomed Diagram (SD1) are presented in Appendix B . These OPLs provide a textual representation that details the processes and state changes of the evolution of a heuristic from a rule of thumb to a principle.

OPM's bimodal property, combining graphical OPD and textual OPL, facilitates a comprehensive representation of complex processes and state changes. The in-zooming mechanism allows for hierarchical decomposition, where processes can be detailed further in subsequent diagrams, enhancing understanding of intricate systems.

3.2 Converting Natural Language to OPM using In-Context Learning

We employ in-context learning to guide the LLM in converting natural language text into OPM models. The process involves providing the LLM with a carefully crafted prompt that includes OPM syntax, semantics, and examples. The prompt details can be found in (Dori and Shteingardt, 2025).

Let T_{NL} be the natural language text (Appendix A) and P_{OPM} the prompt containing OPM instructions and examples. The input to the LLM is:

$$I = P_{\text{OPM}} \circ T_{\text{NL}}, \tag{1}$$

where $\circ$ denotes concatenation. The LLM generates the OPL representation:

$$T_{\text{OPL}} = \text{LLM}(I). \tag{2}$$

This process leverages the LLM's ability to generate structured textual OPL representations from unstructured text, utilizing in-context learning to guide the model's output toward the desired OPL format. The OPL generated by the LLM is presented in Appendix C.

While the preliminary results are encouraging, designing prompts that yield accurate and syntactically correct OPM models from free-form text introduces several challenges. These include pinpointing the primary process, focusing on essential conceptual elements, and clarifying ambiguous relationships in the natural language source. To address these issues, we iteratively refine prompts, adjust instructions, and incorporate carefully chosen examples. Through this iterative approach, the LLM learns to better navigate textual ambiguities and produce more coherent OPM models, thus reducing the need for extensive manual refinement and enabling more reliable neuro-symbolic reasoning pipelines.

3.3 OPM Knowledge-Based Question-Answering System

We developed OPM-QA, an OPM knowledge-based Question-Answering system, that integrates OPM knowledge with the LLM to enhance answer accuracy and reasoning transparency. This system is a core component of NCAI, leveraging the structured knowledge representation of OPM to improve the reasoning capabilities of the LLM.

OPM-QA employs in-context learning by providing the LLM with OPL as the OPM knowledge, a set of example question-answer pairs, and the test questions as context. The knowledge K_{OPL} is derived from the constructed OPM (see Appendix B) and provides a structured and formalized representation. This structured knowledge allows the LLM to reason more effectively when generating answers.

For each test question q_i in the set of test questions Q_{test}, the input to the LLM is formulated as:

$$I_i = K_{\text{OPL}} \circ E_{\text{QA}} \circ q_i, \tag{3}$$

where E_{QA} is the set of example question-answer pairs, and $\circ$ denotes concatenation. The LLM processes this input and generates an answer:

$$a_i = \text{LLM}(I_i). \tag{4}$$

To assess the impact of using structured OPM knowledge on the QA performance, we compare the OPM-QA with a baseline QA system using natural language knowledge (NL-QA). In NL-QA, we replace K_{OPL} with the natural language knowledge K_{NL}, which corresponds to the text provided in Appendix A. This allows us to compare the effectiveness of the structured OPM knowledge against unstructured natural language knowledge in the QA task.

4 Experiments

4.1 Experiment Setup

The purpose of our experiment is to evaluate the effectiveness of the NCAI framework in performing multi-hop reasoning tasks and enhancing reasoning transparency. We aim to compare the performance of OPM-QA with the baseline NL-QA.

Data: We manually developed a dataset of 50 multi-hop reasoning question-answer pairs, following the FanOutQA benchmark (Zhu et al., 2024). These questions are based on the knowledge of the process that transforms informal rules of thumb into well-established principles. The questions require the model to integrate information from multiple statements to arrive at an answer, testing both answer accuracy and reasoning transparency. Examples of the QA pairs are provided in Appendix E, Table 3.

Knowledge Sources: The OPL knowledge K_{OPL} is the OPL generated from the constructed OPM model in Appendix B. The natural language knowledge K_{NL} is the text provided in Appendix A. The QA systems use either K_{OPL} or K_{NL}, along with 5 example QA pairs E_{QA} as context, to answer the 50 test questions Q_{test}.

QA Systems: The QA systems employ in-context learning by providing the LLM with the respective knowledge source, a set of example QA pairs, and the test questions. The LLM used for both systems is `GPT-4o` (version `o1-preview-2024-09-12`), with parameters set to $\textit{temperature} = 0$ and $\textit{top\_p} = 1$ to ensure deterministic output. By using the same LLM and parameter settings, we ensure a fair comparison between OPM-QA and NL-QA. The prompt used in these QA systems is shown in Appendix D. It has been carefully designed to be general enough for QA tasks across various domains and yet instructive enough to guide the model to answer with explicit reference to the OPM elements—processes, objects, and states, thereby increasing reasoning transparency and answer accuracy.

Evaluation Metrics: We evaluate system outputs using a combination of metrics that capture different aspects of answer quality and reasoning transparency. To assess how well the generated answers align with the ground truth in terms of content, we use Loose Accuracy and Strict Accuracy. Loose Accuracy measures the fraction of reference tokens that also appear in the predicted answer after lemmatization, removing stop words, and stripping punctuation, providing a relatively lenient measure of correctness. Strict Accuracy applies a non-linear weighting (with a parameter $k = 1.5$) to penalize partial matches more severely, thus enforcing a stricter standard of correctness.

While Loose Accuracy and Strict Accuracy focus on token-level overlap, ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L) (Lin, 2004) quantify lexical overlap through n-gram and sequence-based comparisons, capturing syntactic similarity between the generated answer and the reference. BLEURT (BT) (Sellam et al., 2020) complements these metrics by providing a more semantic-oriented evaluation, as it uses a learned model to judge the meaning and quality of the generated text. The GPT Judge Score (GPT) (Zhu et al., 2024) further evaluates factual consistency and logical coherence, reflecting how well the answer maintains internal logical structure and correctness from a large language model's perspective.

To address the need for a quantitative measure of reasoning transparency, we propose Transparency Precision ($\mathrm{P_T}$), Transparency Recall ($\mathrm{R_T}$), and Transparency F1 ($\mathrm{F1_T}$). Let $\mathcal{E}_p$ be the set of OPM elements-processes, objects, and states-identified in the prediction, and $\mathcal{E}_g$ the set of OPM elements in the ground truth. Let $\mathcal{E}_{p\cap g}$ be the intersection of these sets, representing correctly matched OPM elements. We define:

$$\mathrm{P_T} = \frac{|\mathcal{E}_{p\cap g}|}{|\mathcal{E}_p|}, \tag{5}$$

$$\mathrm{R_T} = \frac{|\mathcal{E}_{p\cap g}|}{|\mathcal{E}_g|}, \tag{6}$$

$$\mathrm{F1_T} = \frac{2 \cdot \mathrm{P_T} \cdot \mathrm{R_T}}{\mathrm{P_T} + \mathrm{R_T}}. \tag{7}$$

Here, $\mathrm{P_T}$ measures how accurately the predicted reasoning structure identifies the correct OPM elements, $\mathrm{R_T}$ gauges how completely it recovers them, and $\mathrm{F1_T}$ balances both. Together, these transparency metrics provide a statistical measure of how faithfully the system's reasoning aligns with the conceptual logic defined by OPM, offering a principled, quantitative response to calls for more objective assessments of reasoning transparency.

4.2 Results

Table 1 presents the results of our evaluation. For Loose Accuracy, OPM-QA achieves 0.858 ± 0.162, greatly exceeding NL-QA's 0.638 ± 0.212. This indicates that OPM-QA captures a significantly larger fraction of reference tokens under a lenient matching criterion. The difference is statistically significant ($P < 0.001$). Strict Accuracy, which imposes a harsher penalty on partial matches, shows OPM-QA at 0.806 ± 0.213 compared to NL-QA's 0.530 ± 0.252. This improvement is also statistically significant ($P < 0.001$), demonstrating that OPM-QA provides answers that are both more complete and more precisely aligned with the ground truth.

Regarding syntactic overlap measures, OPM-QA significantly outperforms NL-QA in all ROUGE metrics. The ROUGE-1 score for OPM-QA is 0.772 ± 0.159 versus NL-QA's 0.558 ± 0.195, ROUGE-2 is 0.607 ± 0.201 compared to 0.373 ± 0.198, and ROUGE-L is 0.715 ± 0.155 compared to 0.504 ± 0.174. All these differences are highly statistically significant ($P < 0.001$). These results confirm that OPM-QA's generated answers exhibit considerably more lexical and subsequence-level similarity to the reference answers, adhering

Metric	**OPM-QA**	**NL-QA**	**P-value**
Loose Accuracy	$\mathbf{0.858 \pm 0.162}$	0.638 ± 0.212	< 0.001
Strict Accuracy	$\mathbf{0.806 \pm 0.213}$	0.530 ± 0.252	< 0.001
ROUGE-1	$\mathbf{0.772 \pm 0.159}$	0.558 ± 0.195	< 0.001
ROUGE-2	$\mathbf{0.607 \pm 0.201}$	0.373 ± 0.198	< 0.001
ROUGE-L	$\mathbf{0.715 \pm 0.155}$	0.504 ± 0.174	< 0.001
BLEURT	$\mathbf{0.596 \pm 0.165}$	0.474 ± 0.111	< 0.001
GPT Judge Score	$\mathbf{0.920 \pm 0.274}$	0.800 ± 0.404	0.086
Transparency Precision	$\mathbf{0.917 \pm 0.161}$	0.759 ± 0.417	0.015
Transparency Recall	$\mathbf{0.953 \pm 0.143}$	0.455 ± 0.329	< 0.001
Transparency F1	$\mathbf{0.922 \pm 0.136}$	0.546 ± 0.342	< 0.001

Table 1: Evaluation results comparing OPM-QA and NL-QA across correctness, lexical similarity, semantic quality, factual consistency, and transparency. P-values indicate that OPM-QA significantly outperforms NL-QA on all metrics with high statistical confidence, except for GPT Judge and Transparency Precision, where the differences are less significant.

better to the structural and phrasing patterns of the ground truth.

In terms of semantic quality, the BLEURT score for OPM-QA is 0.596 ± 0.165, which surpasses NL-QA's 0.474 ± 0.111. This difference is statistically significant (P < 0.001). This suggests that OPM-QA not only matches lexically but also maintains closer semantic fidelity to the intended meanings of the ground truth answers.

Factual consistency and logical coherence are further evidenced by the GPT Judge Score of 0.920 ± 0.274 for OPM-QA compared to NL-QA's 0.800 ± 0.404. This difference is not statistically significant (P = 0.086), although it still indicates a notable improvement in maintaining factual and logical integrity within the answers.

Most notably, the transparency metrics reveal OPM-QA's substantial advantage in conceptual alignment. OPM-QA achieves a Transparency Precision of 0.917 ± 0.161 and Transparency Recall of 0.953 ± 0.143, whereas NL-QA scores 0.759 ± 0.417 and 0.455 ± 0.329, respectively. The Precision difference is statistically significant (P = 0.015), while Recall remains highly significant (P < 0.001). Consequently, Transparency F1 for OPM-QA is 0.922 ± 0.136 compared to NL-QA's 0.546 ± 0.342, with a P-value of P < 0.001. This metric, which balances Transparency Precision and Transparency Recall, underscores the overall superior performance of OPM-QA in aligning with the ground truth both accurately and comprehensively.

Overall, the majority of these metrics demonstrate statistically significant improvements, affirming the superior performance of OPM-QA over NL-QA. Additionally, the enhancements in Transparency Precision metrics, despite being less statistically significant, further highlight OPM-QA's effectiveness in achieving greater factual consistency and precision in answers. Detailed evaluation results for 10 representative QA examples and additional evaluation tables are provided in Appendix E, including Tables 3, 4, and 5, which further confirm these findings.

4.3 Discussion

The experimental results confirm that grounding the reasoning process in a conceptual model leads to both improved accuracy and clearer interpretability. Compared to its counterpart, the OPM-QA system consistently aligns its reasoning with the well-defined ontology provided by the OPM model. While the NL-QA system may occasionally produce correct or partially correct answers, it often does so without revealing the underlying conceptual structure. In contrast, OPM-QA not only identifies the correct OPM elements-processes, objects, and states-required to transform the heuristic from one state to another but also presents a reasoning chain that is faithful to the conceptual logic defined by OPM.

Table 2 illustrates a representative case where the question focuses on the processes that guide the heuristic from a documented and shared state to a theoretically backed one. The ground truth answer specifies all of the required processes involved in this transformation. OPM-QA's answer successfully enumerates each of these processes, maintaining exact alignment with the conceptual elements

Question	What processes change *Heuristic* from *documented & shared* to *theoretically backed*?
Answer$^{\text{GT}}$	*Heuristic* changes from *documented & shared* to *theoretically backed* through *Testing & Refining*, *Pattern Emerging & Recognizing*, *Effectiveness Validating*, and *Theoretical Backing*.
Answer$^{\text{OPL}}$	The processes that change *Heuristic* from *documented & shared* to *theoretically backed* are *Testing & Refining*, *Pattern Emerging & Recognizing*, *Effectiveness Validating*, and *Theoretical Backing*.
Answer$^{\text{NL}}$	The processes that change *Heuristic* from *documented & shared* to *theoretically backed* are Testing & Refinement, Pattern Recognition, Formal Studies, and *Theoretical Backing*.

Table 2: Comparison of ground truth answer Answer$^{\text{GT}}$ and answers from OPM-QA and NL-QA Answer$^{\text{OPL}}$ and Answer$^{\text{NL}}$ for a sample question, highlighting the matched processes in blue corresponding to the reasoning transparency. While the ground truth and OPM-QA specify all relevant processes, NL-QA mentions only one correct process (*Theoretical Backing*). OPM-QA demonstrates a complete and conceptually aligned reasoning structure, whereas NL-QA's reasoning chain remains incomplete.

defined by the OPM model. In doing so, OPM-QA achieves high transparency metrics, as measured by the previously defined precision, recall, and F1 scores for transparency. Conversely, NL-QA identifies fewer correct conceptual elements, and in some cases introduces extraneous or irrelevant processes. This discrepancy highlights not merely a difference in correctness, but also a fundamental gap in the clarity and coherence of the reasoning steps offered by the two QA systems.

In addition to the tabular comparison, Figure 3 visually confirms that OPM-QA's reasoning pathway closely follows the conceptual map provided by OPM. The figure displays an in-zoomed portion of the OPM model (SD1), where the processes critical to changing the heuristic's state are clearly marked. Each process chosen by OPM-QA is found exactly where it should be according to the conceptual model. Observing these elements in the figure shows that OPM-QA's improved transparency metrics correspond to verifiable reasoning sequences that can be directly traced in the conceptual diagram. This contrasts with NL-QA, whose reasoning cannot be similarly verified, leaving users and experts uncertain of how and why specific processes were mentioned or omitted.

Taken together, these findings demonstrate that integrating conceptual modeling into the QA framework moves beyond improving standard performance metrics. The introduction of quantitative transparency metrics, supported by direct comparisons in both textual and visual forms, underscores how OPM-QA's answers are not just better in terms of correctness, but also clearer, more verifiable, and more trustworthy. This alignment of reasoning with a conceptual backbone is particularly valuable in complex domains where understanding the logic behind an answer is as important as the answer itself. As a result, the synergy between neuro-symbolic reasoning and OPM-based conceptual structures offers a promising avenue toward AI systems that users and domain experts can scrutinize, trust, and ultimately shape with confidence.

5 Conclusion

We propose Neuro-Conceptual Artificial Intelligence (NCAI), a neuro-symbolic approach that integrates OPM conceptual modeling with deep learning to overcome limitations in traditional knowledge representation and reasoning. By embedding OPM-based conceptual logic into a QA system, NCAI captures complex processes and state changes that conventional triplet-based representations and black box neural models struggle to address. Through this structured, bimodal OPM representation, NCAI provides not only improved answer accuracy but also a demonstrably transparent and interpretable reasoning pathway. The introduction of transparency metrics (P_T, R_T, F1_T) offers quantitative support for the alignment with OPM-defined conceptual structures, moving beyond purely qualitative assessments of interpretability.

Our experimental results demonstrate that NCAI substantially outperforms traditional methods on both standard accuracy-based measures and

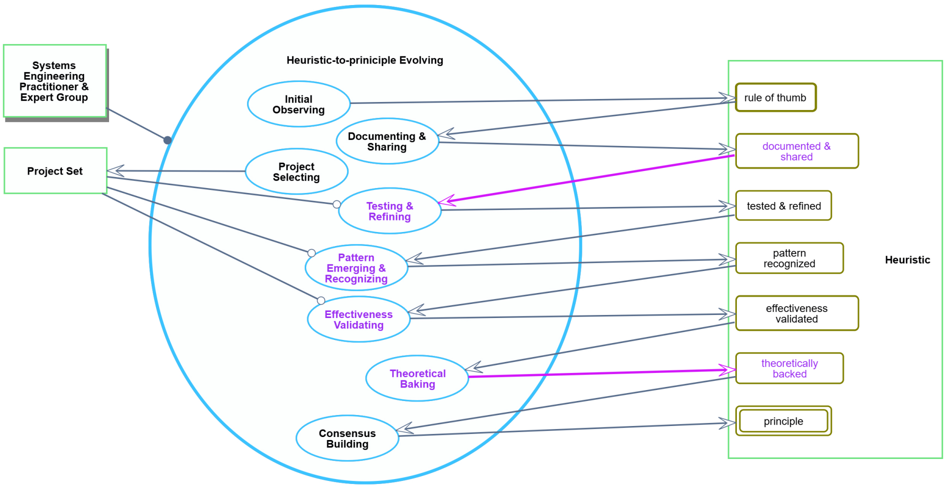


Figure 3: In-Zoomed Diagram (SD1) highlighting the specific processes in blue involved in transforming *Heuristic* from *documented & shared* to *theoretically backed*. These highlighted processes match exactly those identified by OPM-QA in Table 2, demonstrating a coherent and transparent reasoning path.

transparency-focused metrics. By leveraging OPM as a symbolic backbone and employing the LLM under structured guidance, NCAI brings neuro-symbolic AI closer to genuine explainability. Although this work focuses on QA, our conceptual modeling approach may generalize to other tasks requiring robust and interpretable reasoning. Future research will examine scalability to larger, more complex domains, refine prompt designs to handle richer conceptual structures, and integrate our approach with emerging prompting and agentic frameworks. The code and dataset are available on `https://github.com/kangxin/NCAI`.

Limitations

One limitation of our study is that it relies on a relatively small, self-constructed dataset of 50 question-answer pairs. While sufficient for an initial proof of concept, the generalizability and scalability of NCAI to larger and more complex real-world scenarios remain to be explored. In future work, we intend to evaluate NCAI on larger publicly available benchmarks and more intricate conceptual domains, potentially requiring more efficient prompt designs or incremental model updates to handle extensive OPM knowledge.

Additionally, although QA serves as a proof-of-concept task to demonstrate the feasibility of integrating OPM with LLM, applying this approach to other downstream tasks, such as predictive modeling and real-time decision-making in dynamic environments, would require additional domain-specific adaptations and possibly integration with external data sources. While the OPM-based reasoning structure holds promise beyond QA, confirming its utility in these broader contexts remains an area for future investigation.

Moreover, while our method improves transparency through OPM-driven conceptual alignment, certain ambiguities in the source text can still challenge the strict adherence of the LLM to OPM syntax and conventions. The generated OPM representations might require subsequent refinement by human modelers or more specialized training to ensure full syntactic correctness. Developing standardized benchmarks and further metrics for reasoning transparency, as well as exploring more advanced prompting and agentic design patterns, can help refine the approach, but these steps also remain as future endeavors.

Acknowledgments

This research has been supported by the Project of Discretionary Budget of the Dean, Graduate School of Technology, Industrial and Social Sciences, Tokushima University.

References

Thomas R Besold, Artur S d'Avila Garcez, et al. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*.

Davis Brown, Charles Godfrey, Nicholas Konz, Jonathan Tu, and Henry Kvinge. 2023. Understanding the inner-workings of language models through representation dissimilarity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6543–6558, Singapore. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fei Ding, Xin Kang, and Fuji Ren. 2024a. Neuro or symbolic? fine-tuned transformer with unsupervised lda topic clustering for text sentiment analysis. *IEEE Transactions on Affective Computing*, 15(2):493–507.

Fei Ding, Xin Kang, Linhuang Wang, Yunong Wu, Satoshi Nakagawa, and Fuji Ren. 2024b. Causal inference and prefix prompt engineering based on text generation models for financial argument analysis. *Electronics*, 13(9):1746.

Dov Dori. 2002. *Object-Process Methodology: A Holistic Systems Paradigm; with CD-ROM*. Springer Science & Business Media.

Dov Dori, Ahmad Jbara, Natali Levi, and Niva Wengrowicz. 2018. Object-process methodology, opm iso 19450–opcloud and the evolution of opm modeling tools. *Systems Engineering Letters, Project Performance International (PPI) SyEN*, 61:6–17.

Dov Dori and Veronika Shteingardt. 2025. Converting knowledge from text to opm models using generative ai prompts: The neuro-conceptual approach. In *Israel Data Science and AI Initiative 4th Annual Conference*. IDSAI.

Dov Dori et al. 2016. *Model-based systems engineering with OPM and SysML*, volume 15. Springer.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Rongyu Dou and Xin Kang. 2024. TAM-SenticNet: A neuro-symbolic ai approach for early depression detection via social media analysis. *Computers and Electrical Engineering*, 114:109071.

Julia El Zini and Mariette Awad. 2022. Beyond model interpretability: On the faithfulness and adversarial robustness of contrastive textual explanations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1391–1402, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zijian Feng, Hanzhang Zhou, Zixiao Zhu, and Kezhi Mao. 2024. PromptExplainer: Explaining language models through prompt-based learning. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 882–895, St. Julian's, Malta. Association for Computational Linguistics.

Artur d'Avila Garcez and Luis C Lamb. 2023. Neurosymbolic ai: The 3 rd wave. *Artificial Intelligence Review*, 56(11):12387–12406.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

Miriam Horovicz and Roni Goldshmidt. 2024. TokenSHAP: Interpreting large language models with Monte Carlo shapley value estimation. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pages 1–8, Miami, FL, USA. Association for Computational Linguistics.

Mohammadamin Kanaani, Sajjad Dadkhah, and Ali A. Ghorbani. 2024. Triple-R: Automatic reasoning for fact verification using language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16831–16840, Torino, Italia. ELRA and ICCL.

Xin Kang. 2024. Transparency in the dimABSA Task with Neuro-Symbolic and Generative AI. *IEICE Technical Report*, 124(90(CQ2024 16-41)):74–79.

Xin Kang, Zhifei Zhang, Jiazheng Zhou, Yunong Wu, Xuefeng Shi, and Kazuyuki Matsumoto. 2024. TMAK-plus at SIGHAN-2024 dimABSA task: Multi-agent collaboration for transparent and rational sentiment analysis. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 88–95, Bangkok, Thailand. Association for Computational Linguistics.

Hanan Kohen and Dov Dori. 2021. Designing and developing opcloud, an opm-based collaborative software environment, in a mixed academic and industrial setting: An experience report. *Academia Letters*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Edoardo Mosca, Ferenc Szigeti, Stella Tragianni, Daniel Gallagher, and Georg Groh. 2022. SHAP-based explanation methods: A review for NLP interpretability. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4593–4603, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022.

UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546, Seattle, United States. Association for Computational Linguistics.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4149–4158, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eliot Tjoa and Cuntai Guan. 2020. A survey on explainable artificial intelligence (xai): Towards medical xai. *arXiv preprint arXiv:1907.07374*.

Lijie Wang, Yaozong Shen, Shuyuan Peng, Shuai Zhang, Xinyan Xiao, Hao Liu, Hongxuan Tang, Ying Chen, Hua Wu, and Haifeng Wang. 2022. A fine-grained interpretability evaluation benchmark for neural NLP. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 70–84, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States. Association for Computational Linguistics.

Shiquan Yang, Rui Zhang, Sarah Erfani, and Jey Han Lau. 2022. An interpretable neuro-symbolic reasoning framework for task-oriented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4918–4935, Dublin, Ireland. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Hongming Zhang, Yintong Huo, Yanai Elazar, Yangqiu Song, Yoav Goldberg, and Dan Roth. 2023. CIKQA: Learning commonsense inference with a unified knowledge-in-the-loop QA paradigm. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 114–124, Dubrovnik, Croatia. Association for Computational Linguistics.

Miao Zhang, Rufeng Dai, Ming Dong, and Tingting He. 2022. DRLK: Dynamic hierarchical reasoning with language model and knowledge graph for question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5123–5133, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yilun Zhao, Lyuhao Chen, Arman Cohan, and Chen Zhao. 2024. TaPERA: Enhancing faithfulness and interpretability in long-form table QA by content planning and execution-based reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12824–12840, Bangkok, Thailand. Association for Computational Linguistics.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37, Bangkok, Thailand. Association for Computational Linguistics.

A Natural Language Text

Natural Language Text

A plausible set of concise steps of how the process that transforms informal rules of thumb into well-established principles that guide systems engineering practice follows.

1. Initial observation: Heuristics often start as informal rules of thumb based on practical experience.
2. Documentation and sharing: These observations get documented and shared among practitioners.
3. Testing and refinement: The heuristics are tested in various projects and refined based on outcomes.
4. Pattern recognition: As similar heuristics prove useful across multiple projects and domains, recognizable patterns emerge, enabling heuristic generalization.
5. Formal studies: Researchers conduct formal studies to validate the effectiveness of the heuristic.
6. Theoretical backing: The heuristics are connected to underlying theories in systems engineering and related fields.
7. Consensus building: As evidence accumulates, a consensus forms in the systems engineering community about the validity and importance of the heuristic.

B OPL of Constructed OPM Model

OPL for System Diagram (SD)

1. Heuristic can be principle, rule of thumb or at one of five other states. State rule of thumb is initial. State principle is final.
2. Heuristic-to-principle Evolving changes Heuristic from rule of thumb to principle.
3. Systems Engineering Practitioner & Expert Group handles Heuristic-to-principle Evolving.

OPL for In-Zoomed Diagram (SD1)

1. Heuristic-to-principle Evolving from SD zooms in SD1 into Initial Observing, Documenting & Sharing, Project Selecting, Testing & Refining, Pattern Emerging & Recognizing, Effectiveness Validating, Theoretical Baking, and Consensus Building, which occur in that time sequence.
2. Heuristic can be documented & shared, effectiveness validated, pattern recognized, principle, rule of thumb, tested & refined or theoretically backed. State rule of thumb is initial. State principle is final.
3. Systems Engineering Practitioner & Expert Group handles Heuristic-to-principle Evolving.
4. Documenting & Sharing changes Heuristic from rule of thumb to documented & shared.
5. Testing & Refining changes Heuristic from documented & shared to tested & refined.
6. Testing & Refining requires Project Set.
7. Pattern Emerging & Recognizing changes Heuristic from tested & refined to pattern recognized.
8. Pattern Emerging & Recognizing requires Project Set.
9. Effectiveness Validating changes Heuristic from pattern recognized to effectiveness validated.
10. Effectiveness Validating requires Project Set.
11. Theoretical Baking changes Heuristic from effectiveness validated to theoretically backed.
12. Consensus Building changes Heuristic from theoretically backed to principle.
13. Initial Observing changes Heuristic to state rule of thumb.
14. Project Selecting yields Project Set.

C OPL Generated by LLM

OPL Generated by LLM

1. Principle Establishing changes Heuristic from rule of thumb to principle.
2. Practitioner handles Documenting.
3. Practitioner handles Sharing.
4. Researcher handles Formal Studying.
5. Systems Engineering Community handles Consensus Building.
6. Principle Establishing zooms into Observing, Documenting, Sharing, Testing, Refining, Pattern Recognizing, Formal Studying, Connecting, and Consensus Building, which occur in that time sequence.
7. Heuristic can be rule of thumb, documented, shared, tested, refined, recognized pattern, validated, theorized, principle.
8. The state rule of thumb is initial. State principle is final.
9. Observing yields Heuristic.
10. Documenting changes Heuristic from rule of thumb to documented.
11. Sharing changes Heuristic from documented to shared.
12. Testing changes Heuristic from shared to tested.
13. Refining changes Heuristic from tested to refined.
14. Testing requires Project.
15. Refining consumes Outcome.
16. Pattern Recognizing changes Heuristic from refined to recognized pattern.
17. Formal Studying changes Heuristic from recognized pattern to validated.
18. Connecting changes Heuristic from validated to theorized.
19. Connecting consumes Theory.
20. Consensus Building changes Heuristic from theorized to principle.

D Prompt for the QA System

Prompt for OPM-QA and NL-QA Systems

Prompt:

You are a knowledgeable assistant. Your task is to answer questions based on the provided domain knowledge. Your answers should align closely with the domain knowledge, use precise terminology, and remain concise and accurate. Focus on identifying and describing key processes, objects, and states explicitly, and clarify their relationships where relevant.

Domain Knowledge:

[OPL Knowledge in Appendix B or NL Knowledge in Appendix A]

Examples of Question-Answer Pairs:

Q: [example question 1]
A: [example answer 1]
...
Q: [example question N]
A: [example answer N]

New Question:

Q: [question]
A (concise and precise):

E Examples of Questions, Answers, and Evaluation Results

Table 3: 10 example questions and ground truth answers from the QA dataset.

ID	Question	Ground Truth Answer
1	What is the relationship between Testing & Refining and Pattern Emerging & Recognizing in Heuristic evolution?	Testing & Refining changes Heuristic from documented & shared to tested & refined, and Pattern Emerging & Recognizing then changes it from tested & refined to pattern recognized.
2	How does Heuristic achieve theoretical backing before becoming a principle?	Heuristic achieves theoretical backing by undergoing Formal Studying, which changes it from pattern recognized to effectiveness validated, followed by Theoretical Baking, which changes it to theoretically backed, and finally Consensus Building to become a principle.
3	How does Heuristic change from effectiveness validated to principle?	Heuristic changes from effectiveness validated to principle through Theoretical Baking and Consensus Building.
4	How does the Heuristic-to-priniciple Evolving process relate to the different states of Heuristic?	The Heuristic-to-priniciple Evolving process changes Heuristic from rule of thumb through documented & shared, tested & refined, pattern recognized, effectiveness validated, theoretically backed, and finally to principle.
5	What processes change Heuristic from rule of thumb to pattern recognized?	Heuristic changes from rule of thumb to pattern recognized through Documenting & Sharing, Testing & Refining, and Pattern Emerging & Recognizing processes.
6	What processes change Heuristic from rule of thumb to tested & refined?	Heuristic changes from rule of thumb to tested & refined through Documenting & Sharing and Testing & Refining processes.
7	How does Heuristic transition from effectiveness validated to principle?	Heuristic transitions from effectiveness validated to principle through Theoretical Baking and Consensus Building.
8	How does Heuristic transition from documented & shared to pattern recognized?	Heuristic transitions from documented & shared to pattern recognized through Testing & Refining and Pattern Emerging & Recognizing processes.
9	How does Heuristic evolve from documented & shared to effectiveness validated?	Heuristic evolves from documented & shared to effectiveness validated through Testing & Refining, Pattern Emerging & Recognizing, and Formal Studying processes.
10	What processes are involved in Heuristic evolving from rule of thumb to effectiveness validated?	Heuristic evolves from rule of thumb to effectiveness validated through Documenting & Sharing, Testing & Refining, Pattern Emerging & Recognizing, and Formal Studying processes.

Table 4: Evaluation results of OPM-QA for 10 examples. Performance was measured using Loose Accuracy (LA), Strict Accuracy (SA), ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), BLEURT (BT), the GPT Judge Score (GPT), Transparency Precision (P_T) , Transparency Recall (R_T), and Transparency F1 ($F1_T$).

ID	Answer	LA	SA	R-1	R-2	R-L	BT	GPT	P_T	R_T	$F1_T$
1	Testing & Refining precedes Pattern Emerging & Recognizing in Heuristic evolution. Testing & Refining changes Heuristic from documented & shared to tested & refined, while Pattern Emerging & Recognizing subsequently changes Heuristic from tested & refined to pattern recognized. Both processes require a Project Set.	1.000	1.000	0.696	0.591	0.696	0.444	1.000	1.000	1.000	1.000
2	Heuristic achieves theoretical backing through Theoretical Baking, which changes it from effectiveness validated to theoretically backed. Subsequently, it undergoes Consensus Building to transition from theoretically backed to principle.	0.619	0.487	0.714	0.400	0.571	0.496	1.000	1.000	1.000	1.000
3	Heuristic changes from effectiveness validated to principle through Theoretical Baking, which transitions it to theoretically backed, followed by Consensus Building, which transitions it to principle.	1.000	1.000	0.750	0.636	0.750	0.461	1.000	0.800	1.000	0.889
4	The Heuristic-to-principle Evolving process transitions Heuristic through the states of rule of thumb (initial), documented & shared, tested & refined, pattern recognized, effectiveness validated, theoretically backed, and finally to principle (final), following a defined sequence of processes.	0.895	0.846	0.756	0.605	0.756	0.583	1.000	1.000	0.857	0.923
5	The processes that change Heuristic from rule of thumb to pattern recognized are Documenting & Sharing, Testing & Refining, and Pattern Emerging & Recognizing.	1.000	1.000	1.000	0.769	0.857	0.808	1.000	1.000	1.000	1.000
6	The processes that change Heuristic from rule of thumb to tested & refined are Documenting & Sharing followed by Testing & Refining.	1.000	1.000	0.957	0.571	0.783	0.753	1.000	1.000	1.000	1.000
7	Heuristic transitions from effectiveness validated to principle through Theoretical Baking, which changes it to theoretically backed, followed by Consensus Building, which changes it to principle.	1.000	1.000	0.750	0.636	0.750	0.463	1.000	0.800	1.000	0.889
8	Heuristic transitions from documented & shared to pattern recognized through the processes of Testing & Refining, which changes it to tested & refined, followed by Pattern Emerging & Recognizing, which changes it to pattern recognized.	1.000	1.000	0.774	0.621	0.710	0.537	1.000	0.667	1.000	0.800
9	Heuristic evolves from documented & shared to effectiveness validated through the processes of Testing & Refining, Pattern Emerging & Recognizing, and Effectiveness Validating, in that sequence.	0.857	0.794	0.828	0.667	0.759	0.634	1.000	0.750	1.000	0.857
10	The processes involved in Heuristic evolving from rule of thumb to effectiveness validated are Initial Observing, Documenting & Sharing, Testing & Refining, Pattern Emerging & Recognizing, and Effectiveness Validating.	0.812	0.732	0.800	0.667	0.743	0.500	1.000	0.667	1.000	0.800

Table 5: Evaluation results of NL-QA for 10 examples. Performance was measured using Loose Accuracy (LA), Strict Accuracy (SA), ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), BLEURT (BT), the GPT Judge Score (GPT), Transparency Precision (P_T), Transparency Recall (R_T), and Transparency F1 ($F1_T$)

ID	Answer	LA	SA	R-1	R-2	R-L	BT	GPT	P_T	R_T	$F1_T$
1	Testing & Refining provides the practical outcomes and data that enable Pattern Emerging & Recognizing, where recurring success across projects reveals generalizable patterns.	0.417	0.269	0.364	0.194	0.364	0.370	0.000	1.000	0.333	0.500
2	Heuristic achieves theoretical backing through formal studies that validate its effectiveness and by connecting it to underlying theories in systems engineering and related fields.	0.286	0.153	0.421	0.222	0.368	0.428	1.000	0.000	0.000	0.000
3	Heuristic changes from effectiveness validated to principle through Consensus Building, where accumulated evidence and community agreement establish its validity and importance as a guiding principle.	0.778	0.686	0.560	0.435	0.560	0.424	1.000	1.000	0.750	0.857
4	The Heuristic-to-principle Evolving process transitions the heuristic through distinct states: starting as a rule of thumb (Initial Observation), becoming documented and shared (Documentation and Sharing), then tested and refined (Testing and Refinement). It progresses to a pattern recognized state (Pattern Recognition), followed by effectiveness validated (Formal Studies), and finally becomes theoretically backed (Theoretical Backing) and widely accepted through consensus (Consensus Building). Each process step transforms the heuristic into a more formalized and reliable principle.	0.895	0.846	0.493	0.239	0.464	0.404	1.000	0.800	0.571	0.667
5	The processes that change Heuristic from rule of thumb to pattern recognized are Initial Observation, Documentation and Sharing, Testing and Refinement, and Pattern Recognition.	0.692	0.576	0.828	0.519	0.690	0.584	1.000	1.000	0.333	0.500
6	The processes Initial Observation, Documentation and Sharing, and Testing and Refinement change Heuristic from rule of thumb to tested & refined.	0.818	0.740	0.917	0.545	0.417	0.576	1.000	1.000	0.500	0.667
7	Heuristic transitions from effectiveness validated to principle through Theoretical Backing and Consensus Building, where it is connected to underlying theories and gains community agreement on its validity and importance.	0.889	0.838	0.615	0.500	0.615	0.448	1.000	1.000	0.750	0.857
8	Heuristic transitions from documented & shared to pattern recognized through Testing & Refinement, where it is applied across projects, and Pattern Recognition, where recurring usefulness across domains leads to generalization.	0.636	0.508	0.581	0.483	0.581	0.509	1.000	1.000	0.500	0.667
9	Heuristic evolves from documented & shared to effectiveness validated through Testing & Refinement, followed by Pattern Recognition and Formal Studies.	0.643	0.515	0.815	0.640	0.815	0.705	1.000	1.000	0.667	0.800
10	The processes involved in Heuristic evolving from rule of thumb to effectiveness validated are Initial Observation, Documentation and Sharing, Testing and Refinement, Pattern Recognition, and Formal Studies.	0.625	0.494	0.824	0.625	0.765	0.589	1.000	1.000	0.500	0.667